

BÁO CÁO ĐỒ ÁN

ROBUST BOTNET DGA DETECTION: BLENDING
XAI AND OSINT FOR CYBER THREAT
INTELLIGENCE SHARING

Môn học: An toàn mạng - NT140.011.ATCL

Giảng viên: Nghi Hoàng Khoa

Phùng Đức Lương - 21522312

Ngô Minh Quân - 21522492

Chu Nguyễn Hoàng Phương - 21522483

I. Ngữ cảnh

1. Phát triển một mô hình phát hiện lưu lượng DGA của botnet sử dụng các đặc trưng thống kê.
2. Nghiên cứu cách mở rộng sự chia sẻ thông tin về an ninh mạng (CTI) bằng cách sử dụng mô hình AI/ML và các kỹ thuật giải thích trí tuệ nhân tạo (XAI).
3. Nghiên cứu cách cải thiện tính giải thích của các mô hình AI/ML thông qua việc kết hợp XAI và thông tin nguồn mở (OSINT) để tăng cường sự tin tưởng trong quá trình chia sẻ CTI."

II. Phương Pháp

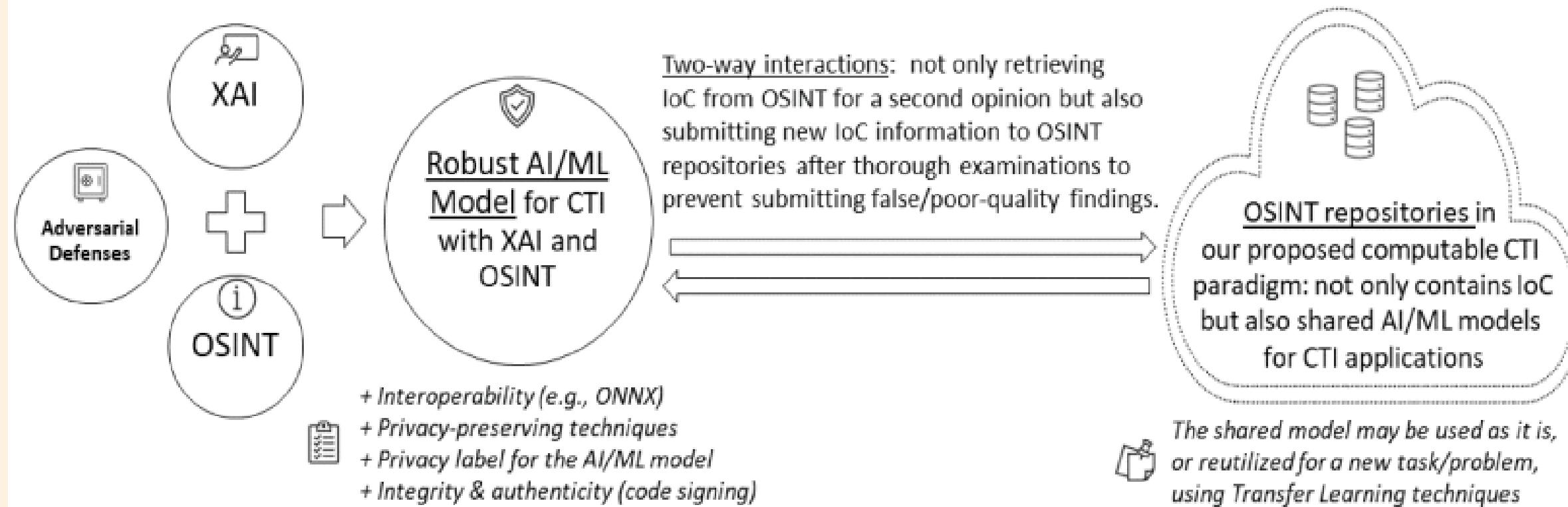
- SHAP sử dụng giá trị Shapley dựa trên lý thuyết trò chơi để cung cấp giải thích toàn cầu về mức độ quan trọng của các đặc trưng.
- LIME sử dụng mô hình thay thế cục bộ để giải thích kết quả phân loại cá nhân của mô hình.
- ANCHORS là một cải tiến của LIME, tạo ra giải thích dễ hiểu dưới dạng các quy tắc IF-THEN.
- Giải thích phản thực tế sử dụng một ví dụ gần nhất để hiển thị thay đổi tối thiểu trong giá trị đặc trưng để đạt được kết quả phân loại khác nhau.

Taxonomy	ANCH-OR	LIME	SH-AP	Counter-factual	XAI-OSINT
<i>Explanation by simplification:</i>					
Rule-based learner	✓	✓	-	-	-
Decision tree	-	-	-	-	-
<i>Feature relevance explanation:</i>					
Influence function	-	-	-	-	-
Sensitivity	-	-	-	-	-
Game Theory inspired	-	-	✓	-	-
Interaction based	-	-	-	-	-
<i>Local explanation:</i>					
Rule-based learner	✓	✓	-	-	-
Decision tree	-	-	-	-	-
<i>Visual explanation:</i>					
Conditional/dependence/ Shapley plots	-	-	✓	-	-
Sensitivity/saliency	-	-	-	-	-
<i>Explanation by example:</i>					
Counterfactual explanation	-	-	-	✓	-
<i>Our proposed approach:</i>					
Second opinion	-	-	-	-	✓

- Ý kiến thứ hai được tạo ra bằng cách kết hợp thông tin từ hai nguồn OSINT (Google Safe Browser và OTX AlienVault). Các truy vấn API được gửi đến các nguồn này để lấy thông tin về tên miền đang bị nghi ngờ.
- Thông tin từ OSINT được kết hợp với kết quả đầu ra của mô hình DGA botnet để tạo ra ý kiến thứ hai.
- Tổng cộng, nghiên cứu này kết hợp các phương pháp XAI và thông tin từ nguồn OSINT để cung cấp giải thích ý kiến thứ hai trong việc phân loại tên miền DGA botnet và hợp pháp.
- Điều này có thể cung cấp sự minh bạch và đáng tin cậy hơn cho quyết định phân loại tên miền trong hệ thống CTI.

Computable CTI Paradigm

Before: security analysts in security operations centers (SOC) have to do **manually** → After: an automated, explainable with OSINT blended (for “second opinion”) approach. Therefore, reducing human intervention in a critical cybersecurity decision-making process.



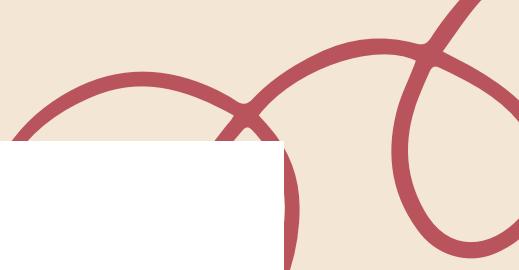
III. Các Mô Hình

- Random Forest
- Naive Bayes
- Logistic Regression
- Extra Tree
- Ensemble Learning

1. Naive Bayes là một thuật toán phân loại dựa trên nguyên tắc của định lý Bayes.
2. Logistic Regression là một thuật toán phân loại dựa trên mô hình hồi quy logistic.
3. Extra Trees là một biến thể của thuật toán Random Forest. Nó sử dụng một tập hợp lớn các cây quyết định ngẫu nhiên để thực hiện phân loại hoặc hồi quy.
4. Random Forest là một thuật toán học máy dựa trên việc xây dựng một tập hợp các cây quyết định ngẫu nhiên.
5. Ensemble Learning là một phương pháp kết hợp nhiều mô hình học máy để tạo ra một dự đoán chung. Ensemble Learning có thể giúp cải thiện độ chính xác và khả năng tổng quát hóa của mô hình.



IV. Bộ dữ liệu thử nghiệm



- Trong nghiên cứu này, ta sử dụng một thuật toán máy học để phát hiện lưu lượng DNS độc hại.
- Để làm điều này, ta cần có dữ liệu thực tế chính xác để huấn luyện mô hình và đánh giá độ chính xác.
- Trong thí nghiệm đầu tiên, ta sử dụng danh sách Alexa Top 1 Million tên miền và 803.333 tên miền thuộc mười botnet DGA families khác nhau.
- Sau đó, ta sử dụng 998.503 tên miền thuộc 55 botnet DGA families từ tài nguyên Netlab 360.
- Sử dụng các tập dữ liệu này, ta có thể huấn luyện mô hình và đánh giá độ chính xác của nó trong việc phát hiện lưu lượng DNS độc hại.

emotet	qijfcnekvhwvcgkg.eu
enviserv	33aaef2199f.net
feodo	mgcdlsidwsdnolwzyz.ru
flubot	oupxbsfpvukowup.cn
fobber	ylbphjjdjs.com
gameover	115vvgdobj3ufljq8gi174q2t7.net
gspy	cc9cf6ae3922d07d.info
kfos	help-google.tw
locky	gcqbsfpkhqf.tf
madmax	www.k3bdsbsa3k.net
matsnu	dishesow-eatcondition.com
mirai	cmbhewcvopvno.online
monerominer	5c95f79304b49.org
murofet	niqlkgqytoirmou.org
mydoom	hrsrrarpst.net
necro	vlxdqiwaifmbashxa.viewdns.net
necurs	hjvtlavpidi.su
ngloweb	subozirion-multirusenlike.com
nymaim	yowgbazj.pw
omexo	8f86373028729e6497f00487d7775f81.net
padcrypt	efddmaenemndoden.website
proslikefan	pbeuiykocu.ru
pykspa	zzzkdgn.org
qadars	7g9ijc9e3why.net
qakbot	pnhwjybkdixb.com

V. Demo

Ta sẽ thêm vào một số thư viện của các thuật toán ML

```
import pandas
import numpy
import matplotlib.pyplot as pyplot
import matplotlib.pylab as pylab
from sklearn.feature_extraction.text import CountVectorizer
import math
from collections import Counter
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import seaborn
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
import joblib
```

#1.Đọc dữ liệu: dataset/data_exported_7features.csv

```
#1 Read The Data: dataset/data_exported_7features.csv
Columns [19] have mixed types. Specify dtype option on import or set low_memory=False.
   Num  No  Domainname  Label  Entropy  RGAlexa  RGCnficker  RGCryptolocker  RGFax  ...  RGZeus  MinRGBotnets  InformationRadius  ClassificationResult  Result  CharLength  LabelBinary  TrueNewFeature  nGramReputation_Alexa
0    0    0  google.com  legit  1.918296  2.255828  2.469318  2.427949  2.475989  ...  2.246443  2.154914  0.824139  legit  Correct  10    0  0.148104  77.921931
1    1    1  facebook.com  legit  2.756986  3.634639  1.847348  1.799824  1.866545  ...  2.396783  3.379829  0.687778  legit  Correct  93    0  0.828927  94.942922
2    2    2  youtube.com  legit  2.921841  2.826693  2.874149  2.684450  2.959482  ...  2.633646  2.084459  0.738282  legit  Correct  31    0  0.353117  88.405453
3    3    3  baidu.com  legit  2.321938  2.153661  2.366616  2.346889  2.364237  ...  2.036576  3.043343  0.785699  legit  Correct  9    0  0.148104  47.568154
4    4    4  yahoo.com  legit  1.921926  2.377799  2.458381  2.509954  2.481281  ...  2.197753  2.328499  0.838762  legit  Correct  9    0  0.148104  55.724565
[5 rows x 24 columns]
```

#2.Tính toán / Chuẩn bị Dữ liệu (Lựa chọn tính năng)

```
#2 Calculate / Prepare The Data (Features Selection)
0                      google.com
1                     facebook.com
2                     youtube.com
3                      baidu.com
4                     yahoo.com
...
1803328      r3o3mt1q7qhld1mp4g2akzqs37.biz
1803329      b05q9rw9lv1d1aq5po08iyjn5.org
1803330      sgem711uuk2vmyl1qlrdymhvl.org
1803331      ozhujl16ayo6crwwdf7fxskdk.org
1803332      9hw0nq1p9binuc6jrifi1noiu.biz
Name: Domainname, Length: 1803333, dtype: object
[[10.          0.14810403 77.92193141  2.15491358]
 [12.          0.82892709 94.94292228  1.37982007]
 [11.          0.3531169  88.40545079  2.00448959]
 ...
 [29.          0.99047121 36.56844303  1.12638625]
 [29.          0.99047121 32.73987616  0.95187191]
 [29.          0.99047121 36.28514263  1.13532149]]
['legit' 'legit' 'legit' ... 'dga' 'dga' 'dga']
[0 0 0 ... 1 1 1]
```

#3 Chuẩn bị dữ liệu thử nghiệm đào tạo

```
... ...
#3 Prepare the trainin testing data
[[1.9000000e+01 3.67645424e-01 4.55180727e+01 8.13191130e-01]
 [1.1000000e+01 2.27128863e-01 5.88216202e+01 1.24281341e+00]
 [1.7000000e+01 9.90471212e-01 8.57920698e+01 1.22478108e+00]
 [1.0000000e+01 2.27128863e-01 5.48440838e+01 1.71007488e+00]
 [1.2000000e+01 1.07095312e-01 6.30734780e+01 1.51538986e+00]
 [1.2000000e+01 8.28927094e-01 5.47573613e+01 1.65765102e+00]
 [1.3000000e+01 3.53116900e-01 7.55837684e+01 1.63917165e+00]
 [2.6000000e+01 6.67704770e-01 1.30233485e+02 6.01193514e-01]
 [1.8000000e+01 2.27128863e-01 8.46203834e+01 1.31512972e+00]
 [9.0000000e+00 3.53116900e-01 3.02428524e+01 1.68727299e+00]
 [1.5000000e+01 1.07095312e-01 9.46712223e+01 1.22795932e+00]
 [9.0000000e+00 1.48104030e-01 3.06942198e+01 2.85861963e+00]
 [1.1000000e+01 1.07095312e-01 6.94123555e+01 1.29555661e+00]
 [1.1000000e+01 8.28927094e-01 3.00239037e+01 1.76826669e+00]
 [3.0000000e+01 9.90471212e-01 2.13129549e+01 6.90144643e-01]
 [1.3000000e+01 8.28927094e-01 6.32349500e+01 1.65623160e+00]
 [3.1000000e+01 1.07095312e-01 2.18043523e+02 8.03720272e-01]
 [1.6000000e+01 1.07095312e-01 7.88883007e+01 1.28793057e+00]
 [1.1000000e+01 8.28927094e-01 2.05092688e+01 1.53032627e+00]
 [2.0000000e+01 8.28927094e-01 4.33338771e+01 5.57401969e-01]
 [3.0000000e+01 9.90471212e-01 2.33894380e+01 6.94688962e-01]
 [1.3000000e+01 2.27128863e-01 8.13628200e+01 1.62068185e+00]
 [1.1000000e+01 3.53116900e-01 7.67967823e+01 1.49888237e+00]
 [1.6000000e+01 1.07095312e-01 1.18747639e+02 1.02562661e+00]
 [1.3000000e+01 8.28927094e-01 5.73139631e+01 1.44587479e+00]
 [1.9000000e+01 3.67645424e-01 1.30777370e+02 7.58992640e-01]
 [3.1000000e+01 8.46070937e-01 1.74774771e+02 4.22102860e-01]
 [2.0000000e+01 8.28927094e-01 3.94990122e+01 5.56425842e-01]
 [1.6000000e+01 7.25975928e-02 9.26554646e+01 1.29961645e+00]
 [2.0000000e+01 9.67922957e-01 5.30795210e+01 4.32389057e-01]
 [1.6000000e+01 2.27128863e-01 5.81990422e+01 1.31861823e+00]
 [1.2000000e+01 1.07095312e-01 5.37870953e+01 1.77170881e+00]
 [9.0000000e+00 1.48104030e-01 4.46163799e+01 1.93815746e+00]
 [1.9000000e+01 9.90471212e-01 2.96357900e+01 1.03391489e+00]]
```

Ta sẽ sử dụng mô hình **RandomForestClassifier** để huấn luyện trên dữ liệu đã được chuẩn bị:

```
#5 RandomForestClassifier
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks | elapsed: 1.7s
[Parallel(n_jobs=40)]: Done 370 tasks | elapsed: 2.0s
[Parallel(n_jobs=40)]: Done 720 tasks | elapsed: 2.4s
[Parallel(n_jobs=40)]: Done 1170 tasks | elapsed: 3.0s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed: 3.4s finished
#6 GaussianNB
```

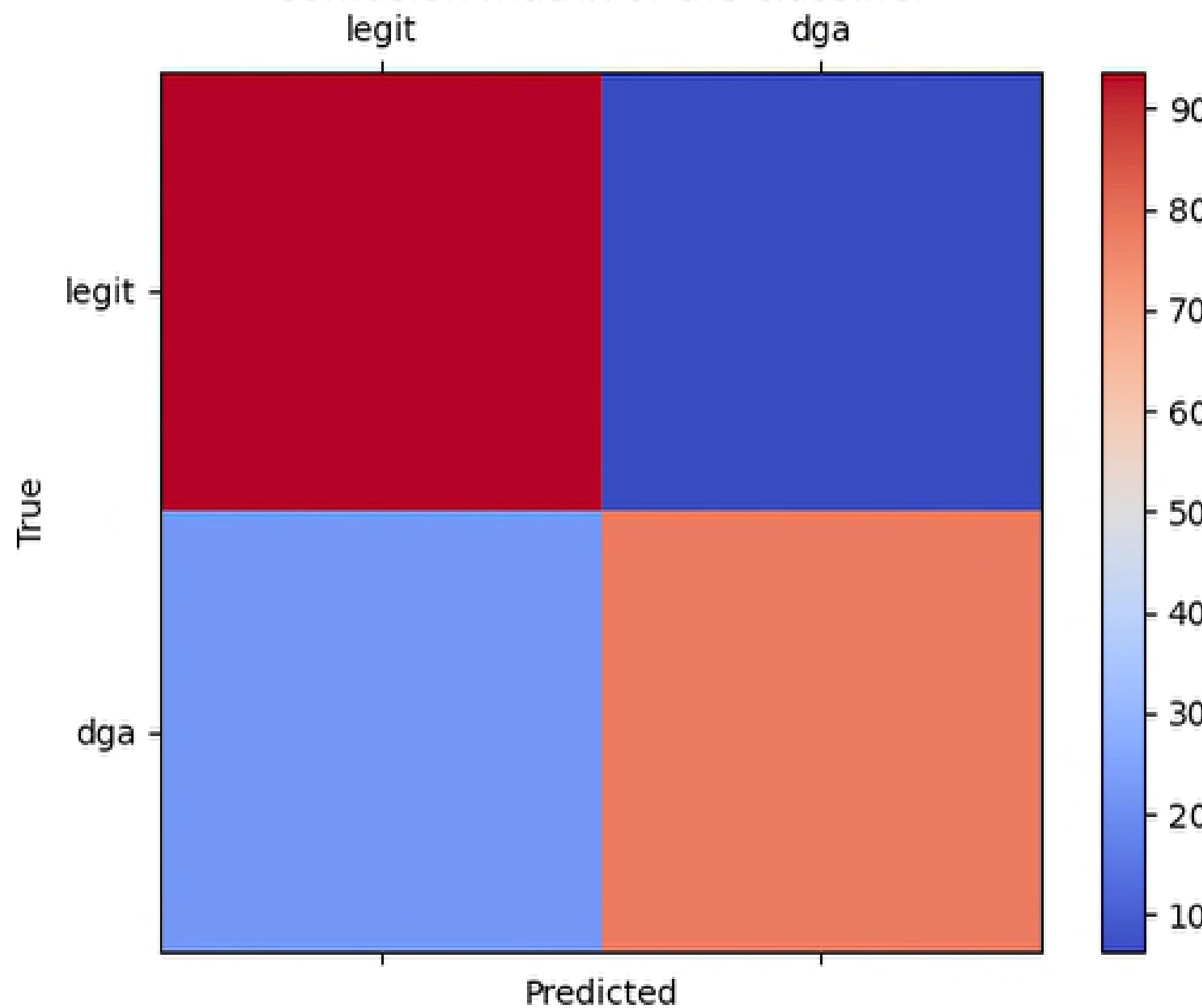
Và sử dụng mô hình **VotingClassifier** để kết hợp các mô hình đã được huấn luyện trước đó:

```
#7 ExtraTreesClassifier
#8 VotingClassifier
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks | elapsed: 0.2s
[Parallel(n_jobs=40)]: Done 370 tasks | elapsed: 0.5s
[Parallel(n_jobs=40)]: Done 720 tasks | elapsed: 1.0s
[Parallel(n_jobs=40)]: Done 1170 tasks | elapsed: 1.6s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed: 1.9s finished
#9 GaussianNB
```

Thực hiện so sánh hiệu suất giữa các mô hình đã huấn luyện sử dụng các phương pháp cross-validation và hiển thị ma trận confusion cho mỗi mô hình:

```
#11 Comparison
Accuracy: 0.960000 (+/- 0.08) [Logistic Regression]
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:   0.2s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:   0.5s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:   0.9s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:   1.4s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:   1.8s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:   0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:   0.2s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:   0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:   0.2s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:   0.5s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:   1.0s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:   1.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:   1.8s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:   0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:   0.1s
```

Confusion matrix of the classifier



LIME, model-agnostic, local explainer

Bước tiếp sử dụng thư viện LIME (Local Interpretable Model-agnostic Explanations) để giải thích dự đoán của mô hình RandomForestClassifier cho một mẫu ngẫu nhiên trong tập kiểm thử:

```
#12 LIME, model-agnostic, local explainer
DOMAIN = twitter.com
CharLength, TreeNewFeature, nGramReputation_Alexa, MinREBotnets
[11. Home 0.14810403 87.39350304 1.6529702 ]
Ground Truth ANSWER = legit
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks | elapsed: 0.0s
[Parallel(n_jobs=40)]: Done 370 tasks | elapsed: 0.1s
[Parallel(n_jobs=40)]: Done 720 tasks | elapsed: 0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks | elapsed: 0.2s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed: 0.2s finished
PREDICTION = legit
CORRECT Prediction :)
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks | elapsed: 0.1s
[Parallel(n_jobs=40)]: Done 370 tasks | elapsed: 0.2s
[Parallel(n_jobs=40)]: Done 720 tasks | elapsed: 0.3s
[Parallel(n_jobs=40)]: Done 1170 tasks | elapsed: 0.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed: 0.6s finished
```

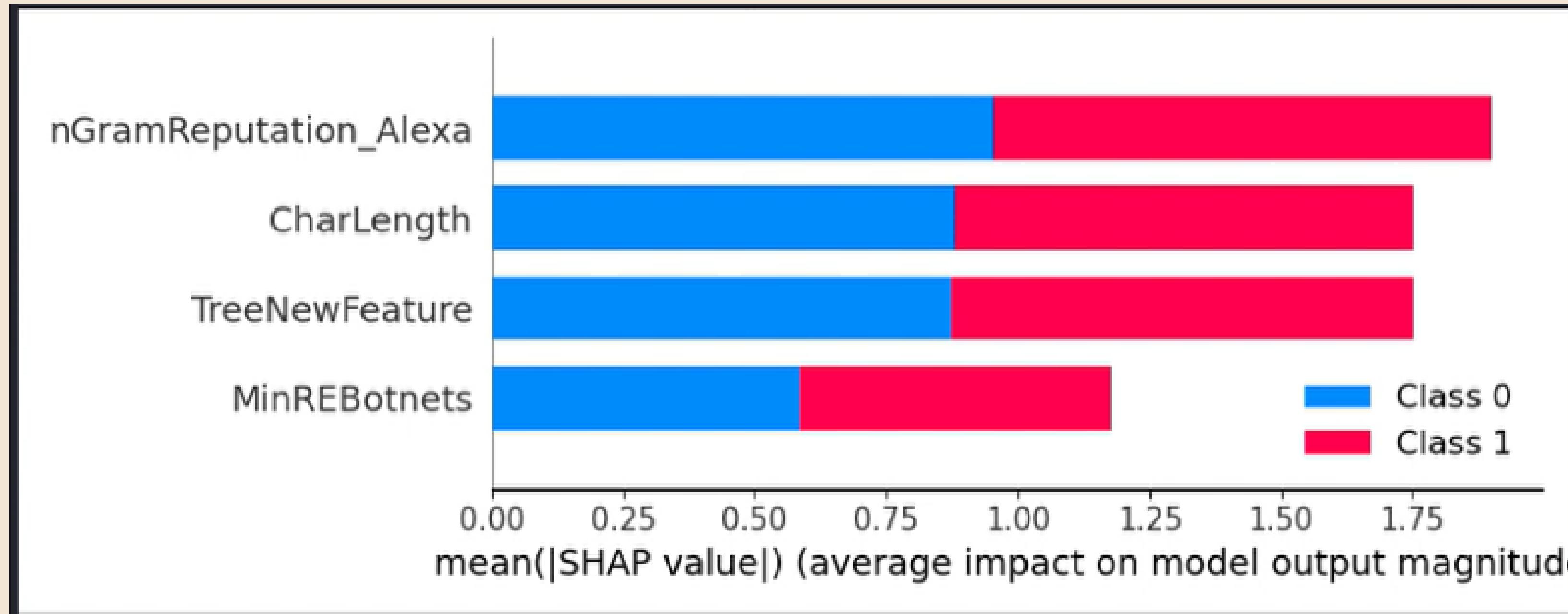
SHAPE, Mô hình bất khả tri với KernelExplainer

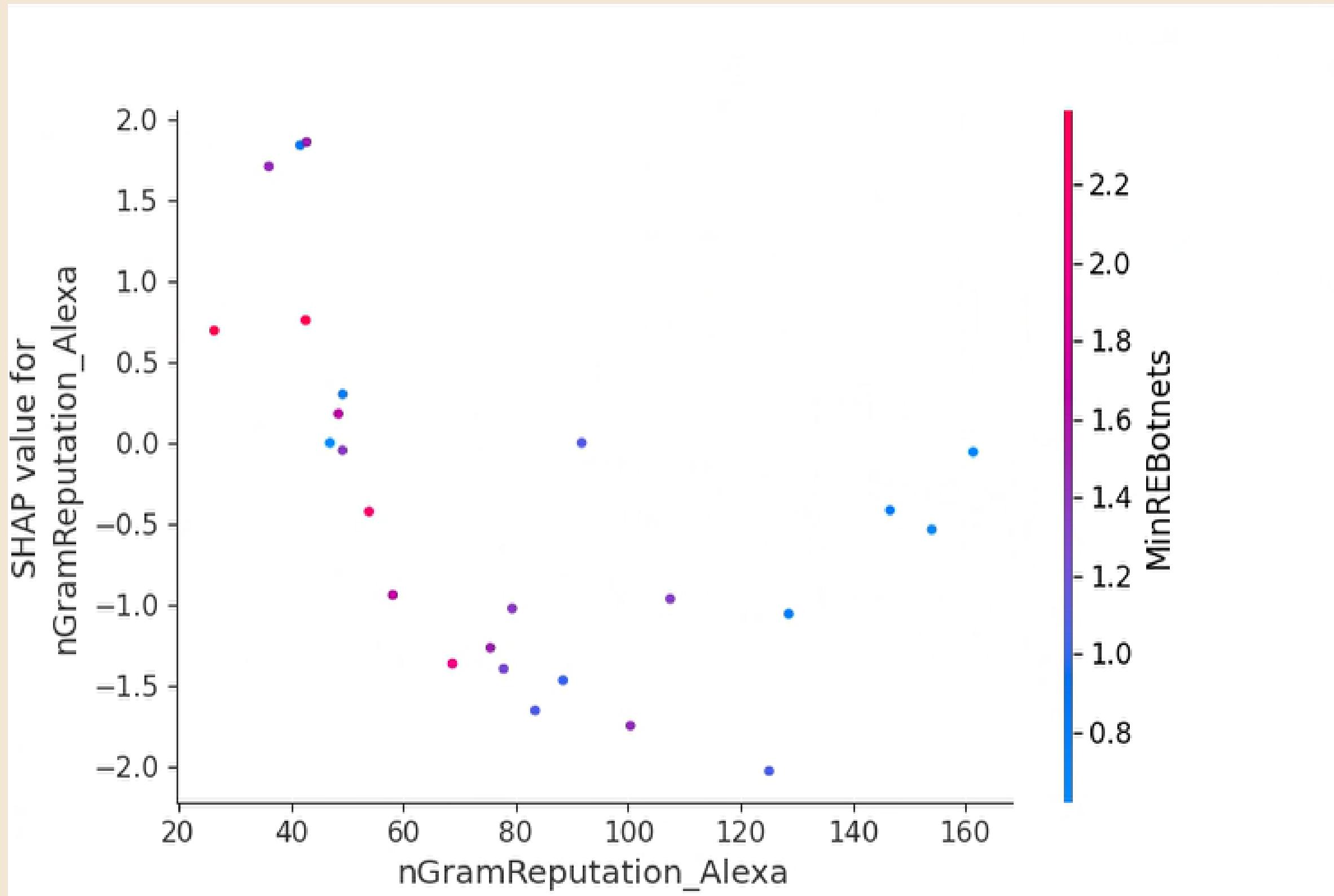
Tiếp đó sử dụng thư viện SHAP (SHapley Additive exPlanations) để giải thích dự đoán của mô hình RandomForestClassifier.

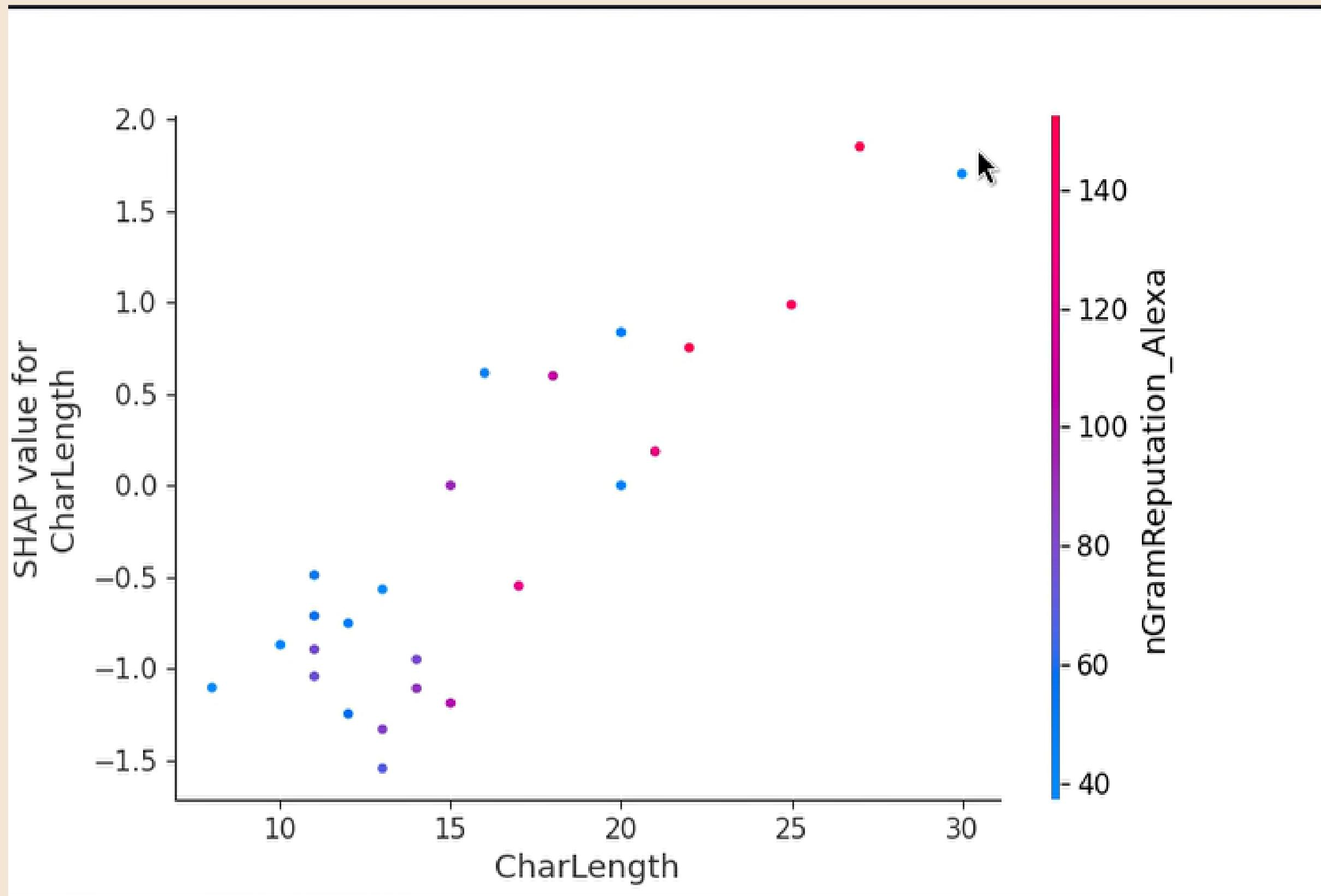
```
#13 SHAPE, Model agnostic with KernelExplainer
<IPython.core.display.HTML object>
    CharLength TreeNewFeature nGramReputation_Alexa MinREBotnets
0          19.0      0.367645           45.518073     0.813191
1          11.0      0.227129           58.821620     1.242813
2          17.0      0.990471           85.792070     1.224781
3          10.0      0.227129           54.844084     1.710075
4          12.0      0.107095           63.073478     1.515390
..        Home ...     ...
95         17.0      0.828927           35.972181     1.394568
96         20.0      0.828927           24.449573     0.556115
97         9.0       0.148104           41.936167     1.958675
98         18.0      0.990471           26.264086     0.811114
99         14.0      0.107095           75.216861     1.073940

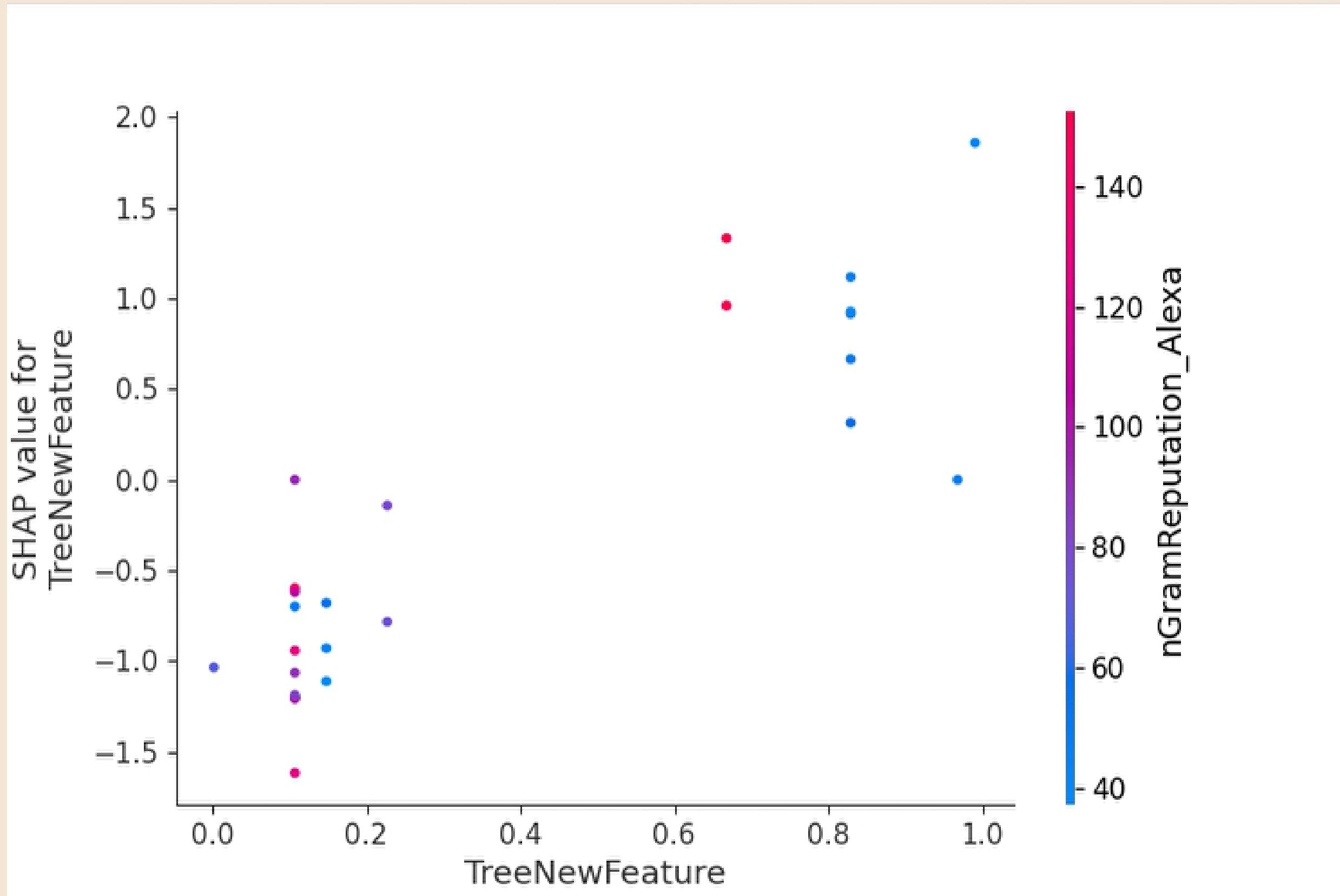
[100 rows x 4 columns]
[[1.1000000e+01 1.07095312e-01 7.78433134e+01 1.25183191e+00]
 [1.2000000e+01 8.28927094e-01 4.84506148e+01 1.67622162e+00]
 [1.4000000e+01 1.07095312e-01 8.84225590e+01 1.03109318e+00]
 [2.1000000e+01 1.07095312e-01 1.28584309e+02 7.37345975e-01]
 [1.0000000e+01 1.48104030e-01 4.25921671e+01 2.43854032e+00]
 [2.0000000e+01 9.67922957e-01 4.69431555e+01 3.13071161e-01]
 [3.0000000e+01 9.90471212e-01 4.16681924e+01 8.37926494e-01]
 [2.2000000e+01 6.67704770e-01 1.54008112e+02 6.02174455e-01]
 [1.2000000e+01 8.28927094e-01 5.80959581e+01 1.69258508e+00]
 [1.3000000e+01 8.28927094e-01 3.60646236e+01 1.44095992e+00]
 [2.7000000e+01 6.67704770e-01 1.61397416e+02 6.87920049e-01]
 [1.4000000e+01 2.27128863e-01 7.93604532e+01 1.35965681e+00]
 [2.5000000e+01 1.07095312e-01 1.46611035e+02 8.06443685e-01]]
```

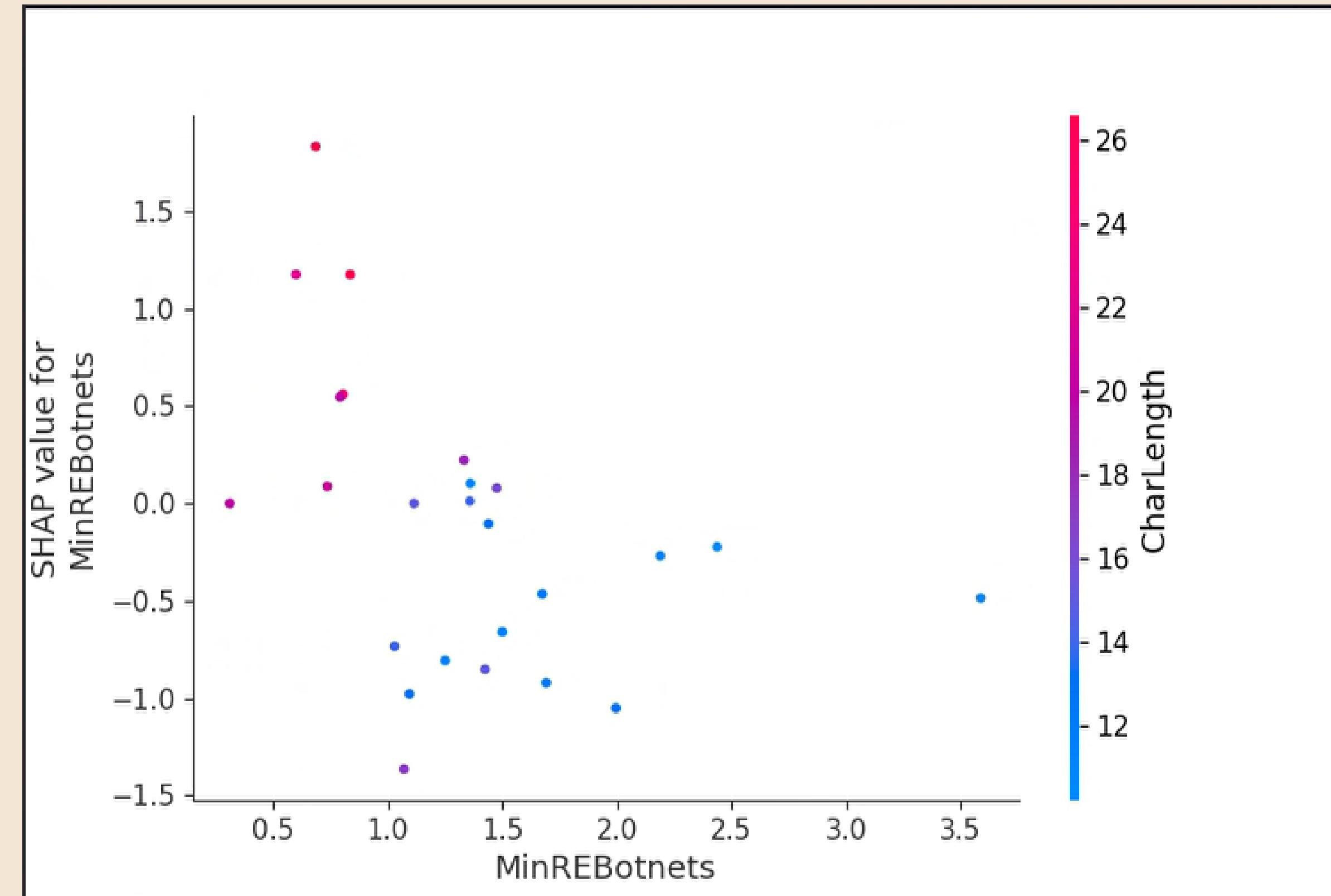
KẾT QUẢ











What If & Giải thích phản thực tế

Chọn ra một tập con nhỏ của dữ liệu từ data để sử dụng trong việc đào tạo và kiểm tra mô hình

#14 What If & Counterfactual Explanations								
		Domainname	CharLength	TreeNewFeature	nGramReputation_Alexa	MinREBotnets	Label	LabelBinary
635829		cadelta.ru	10	0.227129	54.024790	1.254403	legit	0
1345766		joznirne.ru	11	0.828927	36.582503	1.822655	dga	1
1709939		wskakqqikiaskuau.org	20	0.353117	51.984138	0.958705	dga	1
728966	home	nswbar.asn.au	13	0.353117	32.656005	1.739023	legit	0
1429474		invariablythatheother.com	25	0.667705	145.667633	0.665181	dga	1
581107		sbobet.club	11	0.148104	54.302763	1.974802	legit	0
1231022		84bcrusvji1363xriih810b7.net	28	0.990471	33.849807	0.964506	dga	1
914107		endpointvault.com	17	0.107095	100.532460	0.807895	legit	0
708781		myalabama.gov	13	0.148104	62.681412	2.446906	legit	0
1614358		kindtakemusttrainsettlemove.com	31	0.846071	188.369532	0.515225	dga	1
826541		hermosasmujeres.com.co	22	0.107095	155.764247	1.183327	legit	0
923754		countinghouse ltd.com	20	0.667705	135.580883	0.642850	legit	0
265308		realfoodrn.com	14	0.107095	73.265542	1.065933	legit	0
1348529		mangeodi.ru	11	0.227129	28.835311	1.257474	dga	1
1599970		mbrbcfeftlkh.com	16	0.227129	40.845955	1.113502	dga	1
1516306		kjoobqxhyffo.com	16	0.828927	50.173472	1.483576	dga	1
25258		cttv.co	7	0.148104	27.211418	2.881818	legit	0
1401191		britaintobandsshould.com	24	0.667705	138.462025	0.655717	dga	1
1517095		gcbfgbidcbfp.com	16	0.353117	37.317408	1.524242	dga	1
1475188		inheatattemptshistorylaying.com	29	0.846071	180.720023	0.471414	dga	1
209153		echoerom.com	12	0.227129	71.947458	1.434353	legit	0
1110036		dkgwcvdriswxky.co.uk	20	0.990471	70.409569	1.025312	dga	1
1398722		weajosov.ru	11	0.828927	34.627703	1.635959	dga	1
1119740		ohbqxqbvhoyvnc.com	18	0.828927	41.091765	1.424978	dga	1
1707824		emqqgkokcsqqmyge.org	20	0.828927	46.381589	0.675456	dga	1
635829	0							
1345766	1							

Lời giải thích Anchor

Ta sử dụng thư viện alibi để tạo giải thích dự đoán của mô hình thông qua phương pháp Anchor.

```
#14 Anchor explanations
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:   0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:   0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:   0.2s finished
[[1.2000000e+01 8.28927094e-01 2.96258092e+01 1.58428397e+00]
 [2.3000000e+01 3.67645424e-01 1.57749647e+02 8.85284093e-01]
 [1.0000000e+01 1.48104030e-01 2.64345482e+01 2.29821784e+00]
 [3.0000000e+01 9.90471212e-01 1.29650834e+01 7.76746763e-01]
 [1.6000000e+01 3.67645424e-01 1.09764010e+02 9.30723941e-01]
 [3.4000000e+01 4.87624030e-01 2.65811250e+02 5.14045227e-01]
 [3.5000000e+01 6.67704770e-01 2.47318107e+02 6.44233936e-01]
 [1.9000000e+01 8.28927094e-01 3.65073991e+01 1.32202121e+00]
 [1.9000000e+01 2.27128863e-01 1.35782640e+02 1.28814288e+00]
 [3.1000000e+01 9.90471212e-01 1.88408867e+01 9.99848491e-01]
 [1.3000000e+01 1.07095312e-01 9.42525057e+01 1.15317809e+00]
 [1.8000000e+01 1.07095312e-01 1.14553549e+02 1.49502908e+00]
 [1.6000000e+01 1.07095312e-01 1.18766475e+02 8.42756634e-01]
 [1.2000000e+01 8.28927094e-01 5.24167179e+01 1.60626543e+00]
 [1.4000000e+01 3.53116900e-01 9.26822422e+01 1.92645931e+00]
 [1.6000000e+01 8.28927094e-01 9.75231861e+01 1.34434274e+00]
 [1.6000000e+01 3.53116900e-01 3.58861568e+01 1.64865868e+00]
 [2.0000000e+01 9.67922957e-01 5.09561805e+01 3.12746957e-01]
 [1.9000000e+01 1.07095312e-01 1.19136161e+02 9.74141185e-01]
 [2.4000000e+01 1.07095312e-01 1.61102815e+02 1.24489384e+00]
 [2.2000000e+01 1.07095312e-01 1.64218775e+02 9.59589310e-01]
 [2.0000000e+01 8.28927094e-01 3.68852151e+01 8.09217028e-01]
 [6.0000000e+00 1.48104030e-01 1.23764030e+01 2.14551522e+00]
 [1.1000000e+01 2.27128863e-01 3.23615985e+01 1.55264281e+00]
```

Kết quả được giải thích bởi: ANCHOR

```
#15 Result is explained by: ANCHOR
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.2s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.2s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
```

Kết quả được giải thích bởi: LIME

```
#16 Result is explained by: LIME
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.3s
[Parallel(n_jobs=40)]: Done 1170 tasks     | elapsed:    0.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.6s finished
<IPython.core.display.HTML object>
[('CharLength > 20.00', -0.24824565910093352), ('nGramReputation_Alexa > 94.62', 0.19658520859531137), ('TreeNewFeature
{1: [(0, -0.24824565910093352), (2, 0.19658520859531137), (1, 0.19384247785881517), (3, 0.051509030334677484)}]
#17 Rule is explained by: SHAP
```

Kết được giải thích bởi: SHAP

```
#17 RULE is explained by: SHAP
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.0s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks    | elapsed:  0.1s
[Parallel(n_jobs=40)]: Done 370 tasks    | elapsed:  0.2s
[Parallel(n_jobs=40)]: Done 720 tasks    | elapsed:  0.3s
[Parallel(n_jobs=40)]: Done 1170 tasks   | elapsed:  0.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:  0.6s finished
[ 0.2942047 -0.84006309 -0.9533658 -0.63517499]
-0.3635795562274765
```

Kiểm tra bằng API duyệt web an toàn của Google

Ta sẽ thực hiện việc kiểm tra một domain thông qua Google Safe Browsing API để xem liệu domain đó có liên quan đến bất kỳ mối đe dọa nào không

```
-@.3635795562274763
#18 Check by Google Safebrowsing API
http://nicetelecom.us
{'client': {'clientId': 'coba', 'clientVersion': '0.0.1'}, 'threatInfo':
ORMS}], 'threatEntryTypes': ['URL'], 'threatEntries': [{url': 'http://ni
<Response [200]>
No information about this domainname on Google Safe Browser database
#19 Check by OTX Alienvault + ADT
```

```
{'client': {'clientId': 'coba', 'clientVersion': '0.0.1'}, 'threatInfo': {'threatTypes':
['THREAT_TYPE_UNSPECIFIED', 'MALWARE', 'SOCIAL_ENGINEERING',
'UNWANTED_SOFTWARE', 'POTENTIALLY_HARMFUL_APPLICATION'], 'platformTypes':
['ALL_PLATFORMS'], 'threatEntryTypes': ['URL'], 'threatEntries': [{'url':
'http://nicetelecom.us'}]}}
```

Kiểm tra bằng API OTX AlienVault

Tương tự, ta cũng dùng API này để xem liệu domain đó có liên quan đến bất kỳ mối đe dọa nào không

```
No information about this domain name on Google Safe Browsing database
#19 Check by OTX AlienVault API
rghost.net
[  {  'date': '2023-03-24T09:34:21',
    'datetime_int': 1679650461,
    'detections': {  'avast': None,
                    'avg': None,
                    'clamav': None,
                    'msdefender': 'SLFPER:MSIL/AsmbyLoadInvoke'},
    'hash': 'f4f54c91ba0044c130845df2f0baff0ea6ad578bdf2eab5d1074b30201dd4a5a'},
{  'date': '2023-02-02T07:58:12',
    'datetime_int': 1675324692,
    'detections': {  'avast': 'Win32:Small-HTZZ\\ [Trj]',
                    'avg': None,
                    'clamav': None,
                    'msdefender': 'TrojanDownloader:MSIL/Putras.gen!A'},
    'hash': '48c3990f293de4bb40f32f9f4b729d70e11c3fd802d2ff85a5bccf6750cf6379'},
{  'date': '2022-12-24T15:06:50',
    'datetime_int': 1671894410,
    'detections': {  'avast': 'Win32:Evo-gen\\ [Trj]',
                    'avg': None,
                    'clamav': 'Win.Downloader.Jrcx-9759211-0',
                    'msdefender': 'Ransom:MSIL/HiddenTear.TH!MTB'},
    'hash': '0762d584212f180d8f56b17c64b1ae32efad5e23c1cef1e861ed42ea8eb0e981'},
{  'date': '2022-10-06T00:52:37',
    'datetime_int': 1665017557,
    'detections': {  'avast': 'Win32:Evo-gen\\ [Susp]',
                    'avg': None,
                    'clamav': None,
                    'msdefender': None},
    'hash': '3d2ce7aab252997f620d107b4df363b3bedb6c2e65ca903985d74b472b222f8'},
{  'date': '2022-08-07T14:39:44',
    'datetime_int': 1659883184,
    'detections': {  'avast': 'Win32:DropperX-gen\\ [Drp]',
                    'avg': None,
                    'clamav': None,
                    'msdefender': 'TrojanDownloader:MSIL/Minecru.A!bit'},
    'hash': '647b8383b2eba943debe2d36dfbbb8b1201bb0268b54263be29668511cb24bb9'},
{  'date': '2022-08-04T18:33:38',
    'datetime_int': 1659638018,
    'detections': {  'avast': 'Win32:Evo-gen\\ [Susp]',
                    'avg': None,
                    'clamav': 'Win.Malware.Biyb-9754674-0',
                    'msdefender': None},
```

VI. Kết quả thử nghiệm và hạn chế

1. Kết quả

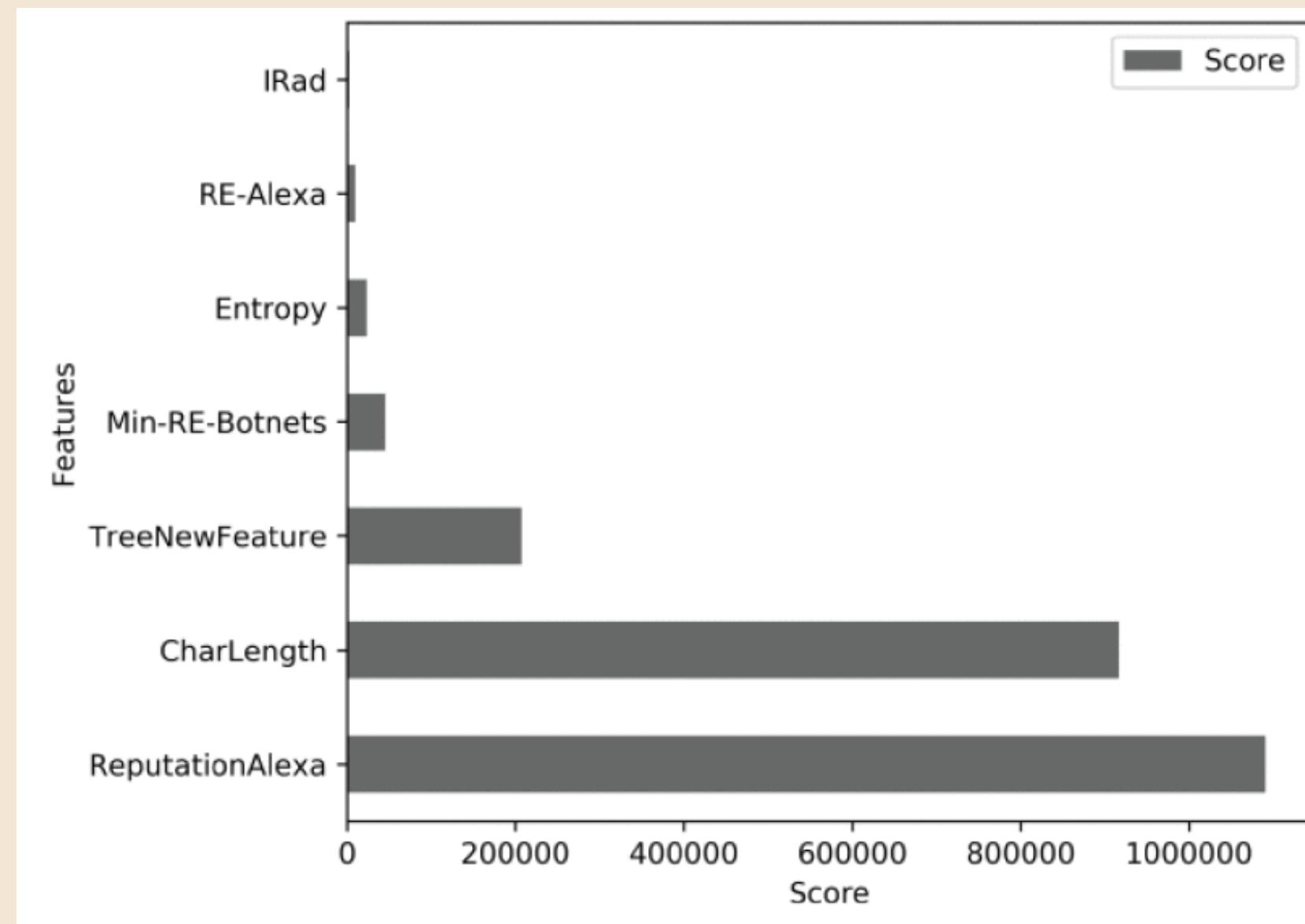
A. So sánh độ chính xác của thuật toán ML

Kết quả của các thí nghiệm của chúng ta được hiển thị trong dưới. Tổng thể, mô hình random forest đạt được độ chính xác cao nhất, tiếp theo là thuật toán extra tree. Và ta có thể thấy rằng naive Bayes luôn cho thấy hiệu suất thấp nhất trong số các thuật toán được so sánh. Độ chính xác cao nhất (96,2%) được đạt được bằng cách sử dụng random forest với tất cả bảy đặc trưng. Ba đặc trưng quan trọng nhất là CharLength, ReputationAlexa và TreeNewFeature.

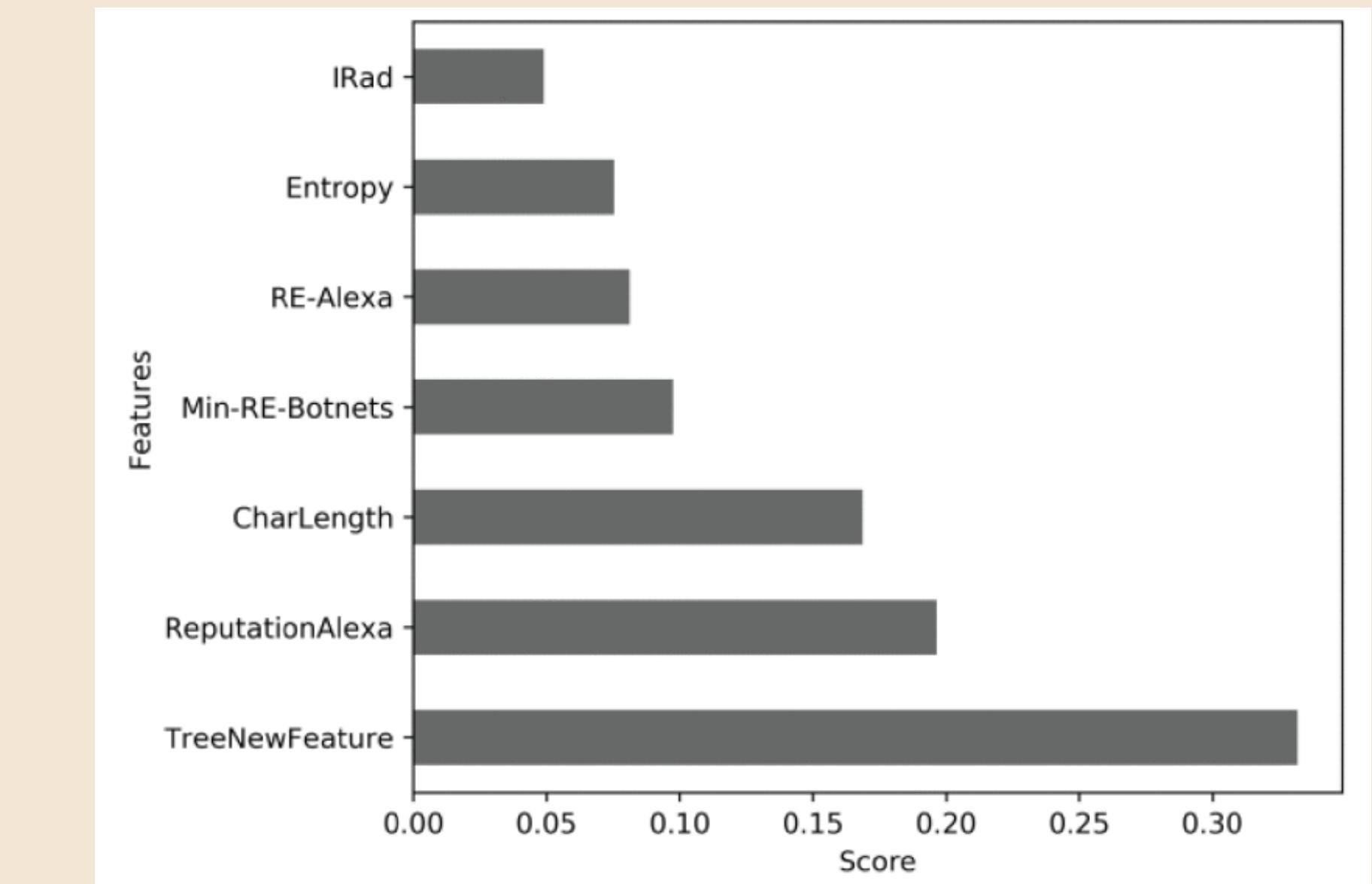
Features	3 features	4 features	4 features	4 features	4 features	5 features	5 features	5 features
- CharLength	✓	✓	✓	✓	✓	✓	✓	✓
- TreeNewFeature	✓	✓	✓	✓	✓	✓	✓	✓
- ReputationAlexa	✓	✓	✓	✓	✓	✓	✓	✓
- RE-Alexa	-	✓	-	-	-	✓	✓	✓
- Min-RE-Botnets	-	-	✓	-	-	✓	-	-
- Entropy	-	-	-	✓	-	-	✓	-
- IRad	-	-	-	-	✓	-	-	✓
Logistic Regression	89.5%	89.8%	89.5%	90.6%	90.0%	90.1%	90.7%	90.0%
Random Forest	92.7%	94.6%	95.1%	94.4%	94.8%	95.7%	95.3%	95.2%
Naive Bayes	83.2%	83.5%	82.7%	82.5%	82.5%	82.9%	82.9%	82.9%
Extra Tree	92.7%	94.3%	94.8%	94.2%	94.5%	95.6%	95.2%	95.1%
Ensemble	91.7%	93.6%	94.4%	94.6%	94.1%	94.7%	94.5%	94.1%

Features	5 features	5 features	5 features	6 features	6 features	6 features	6 features	7 features
- CharLength	✓	✓	✓	✓	✓	✓	✓	✓
- TreeNewFeature	✓	✓	✓	✓	✓	✓	✓	✓
- ReputationAlexa	✓	✓	✓	✓	✓	✓	✓	✓
- RE-Alexa	-	-	-	✓	✓	✓	-	✓
- Min-RE-Botnets	✓	✓	-	✓	✓	-	✓	✓
- Entropy	✓	-	✓	✓	-	✓	✓	✓
- IRad	-	✓	✓	-	✓	✓	✓	✓
Logistic Regression	91.3%	90.9%	90.8%	91.3%	90.5%	90.7%	91.4%	91.3%
Random Forest	95.9%	95.5%	95.6%	96.1%	95.8%	95.8%	96.1%	96.2%
Naive Bayes	81.9%	81.9%	81.9%	82.2%	82.2%	82.3%	81.5%	81.5%
Extra Tree	95.8%	95.4%	95.5%	96.1%	95.7%	95.8%	96.1%	96.2%
Ensemble	95.2%	94.6%	94.8%	95.2%	94.6%	94.6%	95.1%	95.0%

Trong phân tích, ta đã xác định các đặc trưng quan trọng nhất và mối quan hệ của chúng với biến kết quả trong việc phát hiện lưu lượng DNS độc hại. Sử dụng các phương pháp thống kê, đã tìm ra rằng các đặc trưng ReputationAlexa, CharLength và TreeNewFeature có mối quan hệ cao nhất với biến kết quả.



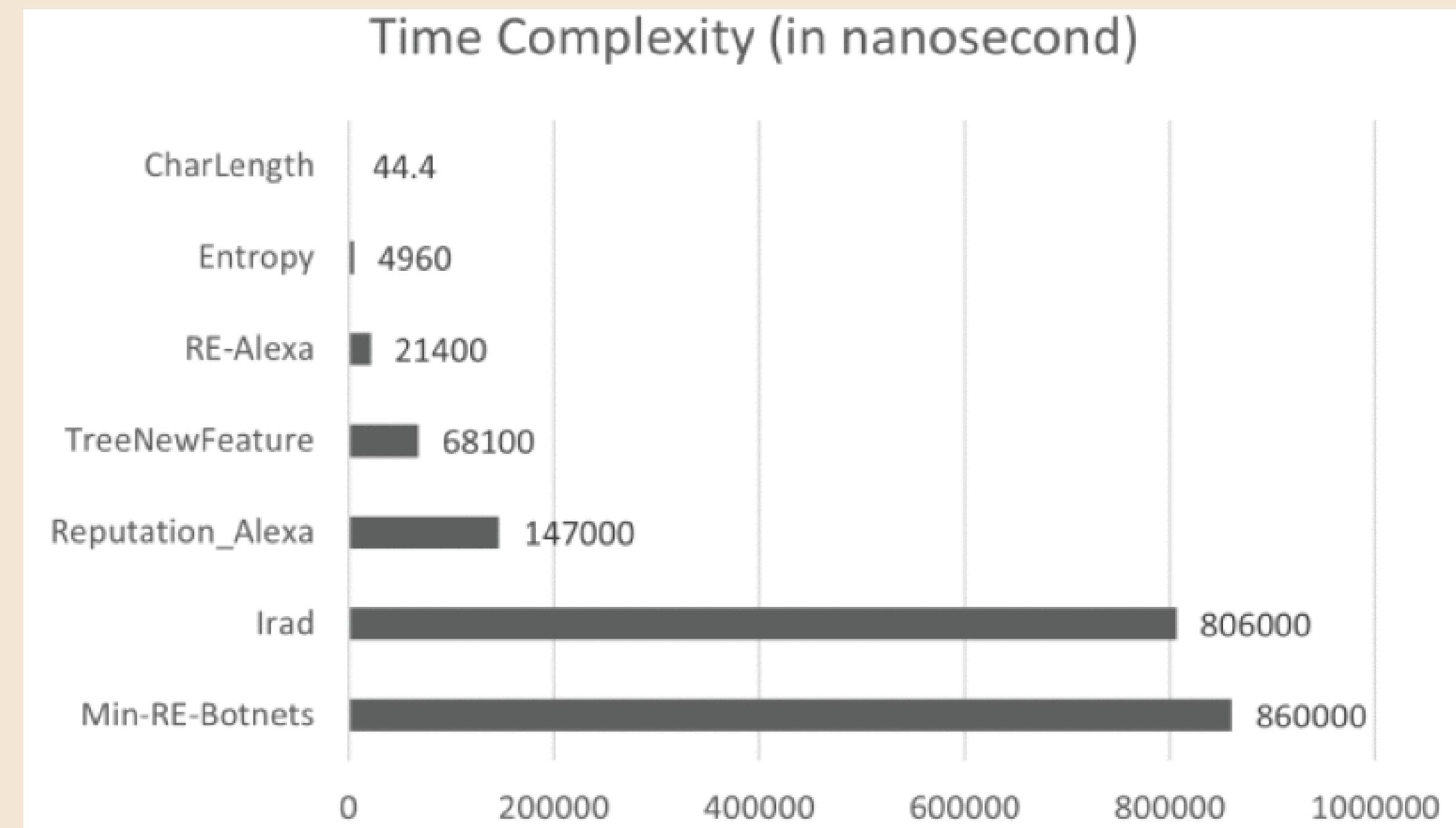
Kết quả kiểm tra chi bình phương.



Kết quả phân tích tầm quan trọng của tính năng

B. Độ phức tạp về thời gian để tính toán các đặc điểm

Kết quả cho thấy rằng các đặc trưng Min-RE-Botnets và IRad yêu cầu thời gian tính toán lâu hơn so với các đặc trưng khác, do độ phức tạp thời gian tuyến tính $O(n)$ của các phương trình liên quan đến số lượng DGA botnet families. Điều này có thể trở thành một nhược điểm khi số lượng DGA tăng lên. Tuy nhiên, đặc trưng ReputationAlexa không đòi hỏi tính toán nặng, vì quá trình chuẩn bị chỉ cần được thực hiện một lần trong quá trình huấn luyện mô hình.



C. So sánh với tác phẩm trước đó

Bảng so sánh mô hình random forest mà chúng tôi đề xuất với công trình trước đây sử dụng cùng phương pháp và thuật toán. Kết quả cho thấy mô hình của chúng ta đạt được tỷ lệ phát hiện tốt hơn với độ chính xác trung bình là 98,9%, mặc dù chỉ sử dụng bảy đặc trưng so với 24 đặc trưng của công trình trước đó.

Datasets	Our Random Forest Model (with only 7 features)	Hoang and Vu [7] (with 24 features)
39 DGA families	99.3%	83.8%
25 DGA families	99.7%	98.0%
10 DGA families	97.2%	83.1%
4 DGA families	99.3%	75.0%
<i>average =</i>	<i>98.9%</i>	<i>85.0%</i>

D. Đánh giá độ bền

Đầu tiên, chúng ta kiểm tra hiệu suất của mô hình random forest của chúng ta với bảy đặc trưng bằng cách sử dụng tập dữ liệu ground-truth gồm tên miền của Alexa và 55 DGA families. Mô hình của chúng ta đạt được độ chính xác 96,3%.

Tuy nhiên, khi đánh giá tính ổn định chống lại các cuộc tấn công CharBot, MaskDGA và DeepDGA, mô hình của chúng tôi cho thấy khả năng phòng vệ tốt hơn đối với cả ba cuộc tấn công DGA.

Datasets	OUR MODEL	Character-Based Deep Learning Model [8]			
		Endgame [35]	CMU [36]	NYU [37]	MIT [38]
- Alexa Top 1M and 55 DGA families	96.3%	98.9%	99.0%	98.9%	99.0%
- CharBot, MaskDGA, and DeepDGA attacks	44.2%	35.8%	30.5%	38.4%	29.7%
- CharBot attack	17.4%	15.0%	10.9%	9.1%	10.0%
- MaskDGA attack	19.7%	27.4%	17.1%	16.3%	30.9%
- DeepDGA attack	45.8%	36.7%	31.6%	40.0%	30.3%

2. Hạn chế

Nhược điểm của mô hình phát hiện DGA của chúng ta là độ phức tạp thời gian khi tính toán các đặc trưng và độ mạnh mẽ hạn chế đối với các cuộc tấn công MaskDGA. Cải tiến trong tương lai nên tập trung vào việc xây dựng các đặc trưng tốt hơn và các chiến lược phòng thủ chống lại cuộc tấn công.

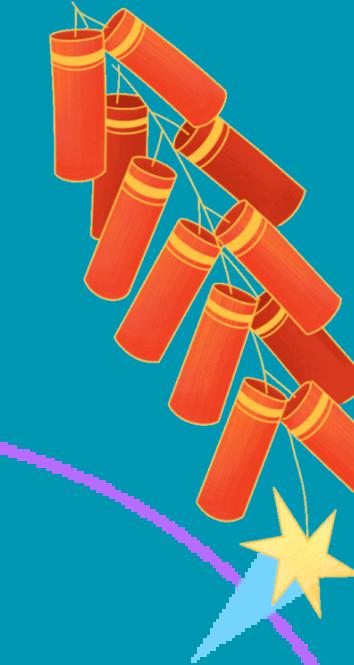
VII. Công việc tương lai

Cải tiến trong tương lai nên tập trung vào việc tạo ra các đặc điểm tốt hơn và chiến lược phòng thủ chống đối địch. Phòng thủ mục tiêu di động (MTD) có thể tăng cường tính mạnh mẽ của mô hình bằng cách kết hợp các mô hình khác nhau để hoạt động cùng nhau.

REFERENCE:

H. Suryotrisongko, Y. Musashi, A. Tsuneda and K. Sugitani, "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing," in IEEE Access, vol. 10, pp. 34613-34624, 2022, doi: 10.1109/ACCESS.2022.3162588.

THANK
YOU!



CẢM ƠN MỌI NGƯỜI ĐÃ CHÚ Ý LẮNG NGHE

